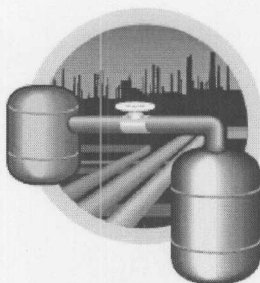


Process Controllers

The Superposition Principle allows feedback controllers to anticipate how linear processes will react to control efforts.

Vance J. VanDoren,
Ph.D., P.E.
Control Engineering



A feedback controller can steer a process variable toward the desired set-point only if it can somehow predict the future effects of its current control efforts. A model-based controller does so with the help of a mathematical representation of the process's behavior. A well-tuned PID loop uses an implicit model characterized by the values of the controller's P (proportional), I (integral), and D (derivative) parameters.

These two techniques compute their control efforts differently, but they both rely on the *linearity* of the process to anticipate how it is going to respond. A process is said to be *linear* if the process variable increases by a factor of u when the control effort is increased by the same amount. And if two separate sequences of control efforts are added together and applied to a linear process, the resulting values of the process variable will always equal the sum of the values that would have resulted had the two control efforts been applied separately.

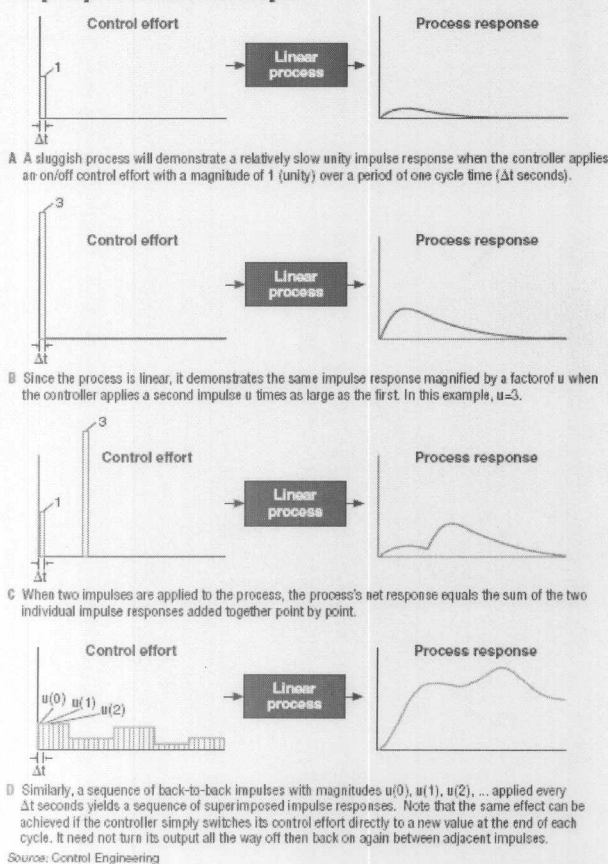
Superposition

This predictability gives rise to the *Superposition Principle* which governs the behavior of all linear processes. The "Superposition Principle" graphic shows how it works in four situations where a computer-based controller with a cycle time of Δt

seconds has applied a different sequence of control efforts to the same linear process.

In case A, the controller has applied a single *impulse* with a magnitude of 1 unit (percent, degree, PSI, whatever) and a width of one cycle time (Δt seconds). The resulting fluctuations in the process variable are known as the process's *impulse response* or, in this particular case, its

Superposition Principle



ONLINE

For more tutorials on control loops, loop tuning and PID, search "Vance" atop www.controleng.com

Predict the Future

unity impulse response.

The process in this example happens to be somewhat sluggish, so its unity impulse response rises and falls relatively slowly as the effects of the impulse wear off. This could represent any number of industrial processes, such as the temperature in a vat after a heating element has been turned on then off again, or the flow rate in a pipe after a valve has been opened then closed.

Case B shows how increasing the magnitude of the impulse increases the magnitude of the impulse response but not its general shape. The second impulse is three times as large as the first, so the magnitude of the impulse response has been tripled.

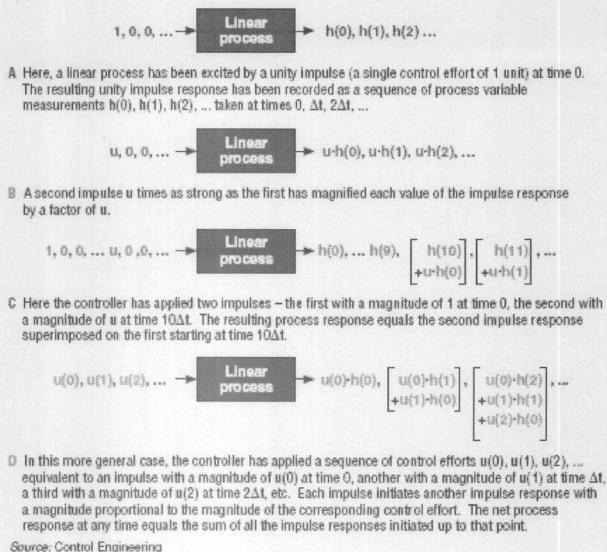
In case C, both impulses have been applied to the process, but at different times. The process's net response after the second impulse equals the sum of the two impulse responses added together point by point. The second impulse response has been effectively "superimposed" on the first, hence the name of the principle that describes this phenomenon.

Case D shows that a contiguous sequence of impulses with magnitudes of $u(0)$, $u(1)$, $u(2)$, ... applied to the process at times 0 , Δt , $2\Delta t$, ... has the same additive effect. Each new impulse response adds to the impulse responses already in progress, and the magnitude of each is determined by the magnitude of the impulse that caused it. The process's net response at any time is the sum of all the impulse responses that have been initiated up to that point.

Equivalent calculation

Thanks to the Superposition Principle, a controller can predict how a linear process will respond to any sequence of control efforts, not just impulses. It also gives an algorithm for computing the resulting values of the process variable, as shown in "Calculating the Process Response."

Calculating the Process Response



This graphic depicts the same four situations, except that the control efforts and the corresponding process responses are represented by their numerical values rather than trend charts. Each data stream has been sampled and recorded once every Δt seconds, hence the expression *sampling interval* often used to describe the controller's cycle time Δt .

Case D shows the calculations required to compute the values of the process variable $y(0)$, $y(1)$, $y(2)$, ... that would result from an arbitrary sequence of control efforts $u(0)$, $u(1)$, $u(2)$, ... Specifically,

$$\begin{aligned} y(0) &= u(0) \cdot h(0) \\ y(1) &= u(0) \cdot h(1) + u(1) \cdot h(0) \\ y(2) &= u(0) \cdot h(2) + u(1) \cdot h(1) + u(2) \cdot h(0) \end{aligned}$$

etc. Each calculation gets successively longer as more and more impulses figure into the result. Fortunately, there's a convenient way to organize all these multiplication and addition operations, as shown in the "Convolution" table, where two infinitely long "numbers"

PROCESS CONTROL

Convolution

	Column 1	Column 2	Column 3	Column 4	...
	$h(0)$	$h(1)$	$h(2)$	$h(3)$	...
	$u(0)$	$u(1)$	$u(2)$	$u(3)$	...
Row 1	$u(0) \cdot h(0)$	$u(0) \cdot h(1)$	$u(0) \cdot h(2)$	$u(0) \cdot h(3)$	...
Row 2		$u(1) \cdot h(0)$	$u(1) \cdot h(1)$	$u(1) \cdot h(2)$	...
Row 3			$u(2) \cdot h(0)$	$u(2) \cdot h(1)$	...
Row 4				$u(3) \cdot h(0)$	...
...					...
	$y(0)$	$y(1)$	$y(2)$	$y(3)$	...

This table organizes the calculations required to compute the values of the process variable $y(0), y(1), y(2), \dots$ that will result when control efforts $u(0), u(1), u(2), \dots$ are applied to a linear process with a unity impulse response of $h(0), h(1), h(2), \dots$. Row 1 is calculated by multiplying each value of the unity impulse response by $u(0)$. This represents the impulse response initiated by the first control effort $u(0)$. Row 2 is calculated the same way except that the results are recorded starting in column 2 rather than column 1, and the multiplier is $u(1)$. This represents the second impulse response initiated by control effort $u(1)$. Each subsequent row is calculated similarly, starting one column further to the right and using the next control effort as the multiplier. Adding the entries in each column from row 1 down to row $k-1$ sums all of the impulse responses initiated from time 0 through time $k-1$ and thereby yields the values of $y(0)$ through $y(k)$. The value of $y(k)$ can also be calculated individually using the convolution sum:

$$y(k) = u(0) \cdot h(k) + u(1) \cdot h(k-1) + \dots + u(k-1) \cdot h(1) + u(k) \cdot h(0)$$

For example, the table's fourth column show that

$$y(3) = u(0) \cdot h(3) + u(1) \cdot h(2) + u(2) \cdot h(1) + u(3) \cdot h(0)$$

Source: Control Engineering

$$H = h(0), h(1), h(2), \dots$$

and

$$U = u(0), u(1), u(2), \dots$$

are "multiplied" together to compute

$$Y = y(0), y(1), y(2), \dots$$

using the familiar long multiplication algorithm, but with data points $h(0), h(1), h(2), \dots$ and $u(0), u(1), u(2), \dots$ instead of individual digits. This calculation, known as *convolution*, is actually the mirror image of long multiplication. The multiplication and addition steps are the same, but it does not involve any carry-over from one column to the next. It is typically written as $Y = H * U$ where $*$ is the *convolution operator*.

Convolution is the basis for an entire mathematical discipline known as *linear systems analysis*. It gives control engineers a powerful tool for analyzing the behavior of linear processes and designing feedback controllers that can predict the future.

Vance Van Doren, Ph.D., P.E., is senior editor for Control Engineering. He can be reached at vance@control.com.

mCon 3000

People | Power | Partnership

www.HARTING-USA.com

HARTING mCon 3000 managed Ethernet Switch - solutions for reliable operation of Industrial Ethernet networks

The HARTING mCon 3000 line of Ethernet Switches offers the full suite of management functions needed in convergent networks and administration from a central location via SNMP.

IGMP snooping functionality ensures Ethernet/IP capability.

VLAN allows better communication flow and avoids unnecessary network loads by using virtual segmentation.

RSTP redundant networks are arranged for vital network communication continuation after component, cable or connection failure.

Fiber optic ports available.

HARTING North America | 1370 Bowes Road | Elgin, IL 60123 | 877.741.1500 | more.info@HARTING.com

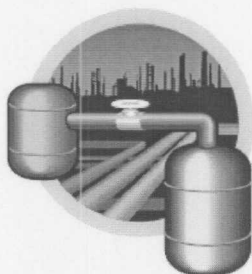


PROCESS CONTROL

Analyzing Closed-Loop Behavior with Convolution

The second in a two-part series explaining how linear process behavior can be predicted mathematically.

Vance VanDoren,
Ph.D., P.E.
Control Engineering



The repeatable behavior of a linear process allows a process controller to predict the future effects of its current efforts. As described in "Process Controllers Predict the Future" (March 2008), a controller's predictive abilities can be reduced to a mathematical formula by conducting the following experiment:

1) Stimulate the process in open-loop mode by forcing the controller to apply a *unity impulse*; that is, a single control effort that is one unit in magnitude and Δt seconds in duration.

2) Determine the process's open-loop *impulse response* by measuring the process variable every Δt seconds after the impulse.

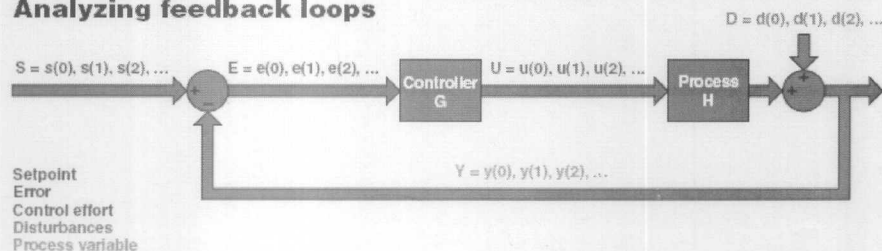
3) Label those measurements in chronological order as $h(0)$, $h(1)$, $h(2)$, etc. and construct a single infinitely long "number" H using those values as "digits". That is, let

$$H = h(0), h(1), h(2), \dots$$

4) Start over and let the controller apply an arbitrary sequence of control efforts to the process. Label those values $u(0)$, $u(1)$, $u(2)$, etc., and use them to construct a second infinitely long number U :

$$U = u(0), u(1), u(2), \dots$$

Analyzing feedback loops



Source: Control Engineering

The behavior of this closed-loop control system can be predicted by constructing four "numbers"— S , D , G , and H —with "digits" consisting of sampled values of the setpoint, the disturbance, the open-loop impulse response of the process, and the open-loop impulse response of the controller, respectively. The sequence of process variable measurements Y that will result from any given combination of S , D , G , and H can then be computed from

$$Y = \frac{G * H}{1 + G * H} * S + \frac{1}{1 + G * H} * D$$

where the star (*) indicates convolution and the two quotients indicate deconvolution.

Source: Control Engineering

ONLINE

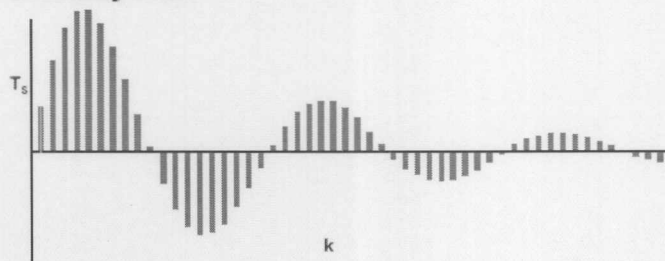
Go to www.controleng.com/archive, to find more information on loop tuning and advanced control, including:

"Process Controllers Predict the Future," March 2008

"Frequency Domain Analysis Explained," October 2005

PROCESS CONTROL

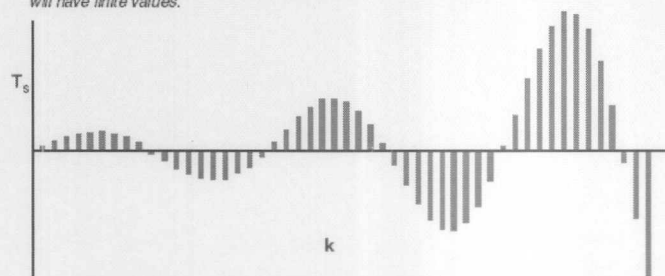
Stability test



This histogram shows each of the digits calculated for the transfer function

$$T_s = \frac{G \cdot H}{1 + G \cdot H}$$

that corresponds to one particular closed-loop system composed of a controller with an open-loop impulse response of G and a process with an open-loop impulse response of H . The graph of T_s shows how the process variable would react if the closed-loop system were to be stimulated with a single unity impulse (that is, if the setpoint sequence were to be changed to $S = 1, 0, 0, \dots$). The shape of this graph will vary for different combinations of processes and controllers, but if the closed-loop system is stable, all of the digits of T_s will have finite values.



For an unstable closed-loop system, the histogram of T_s will not turn out to be entirely finite, and the system's closed-loop impulse response will grow larger and larger over time.

Source: Control Engineering

5) Compute $Y = H * U$ using *convolution*, which multiplies H and U together just as if they were normal multi-digit numbers. That is, use normal multiplication and addition to compute

$$y(0) = u(0) \cdot h(0)$$

$$y(1) = u(0) \cdot h(1) + u(1) \cdot h(0)$$

$$y(2) = u(0) \cdot h(2) + u(1) \cdot h(1) + u(2) \cdot h(0)$$

and so on until the k th sampling interval when

$$y(k) = u(0) \cdot h(k) + u(1) \cdot h(k-1) + \dots + u(k-1) \cdot h(1) + u(k) \cdot h(0)$$

If the process is truly linear and there are no other influences affecting the process variable, then these computed values should match the

actual process variable measurements that will eventually result from the control efforts U .

Conversely, the equations in step 5 could be solved for the open-loop control efforts U that would be necessary to generate a specified sequence of process variable measurements Y . That is, pick the desired values of

$$Y = y(0), y(1), y(2), \dots$$

and let

$$u(0) = \frac{y(0)}{h(0)}$$

$$u(1) = \frac{1}{h(0)} [y(1) - u(0) \cdot h(1)]$$

$$u(2) = \frac{1}{h(0)} [y(2) - u(0) \cdot h(2) - u(1) \cdot h(1)]$$

and so on until the k th sampling interval when

$$u(k) = \frac{1}{h(0)} [y(k) - u(0) \cdot h(k) - u(1) \cdot h(k-1) - \dots - u(k-1) \cdot h(1)]$$

This operation, known appropriately enough as *deconvolution*, is essentially the long division algorithm applied to dividing Y by H to get $U = Y/H$. See the "Deconvolution example" graphic.

Closed-loop analysis

In practice, process controllers typically compute their control efforts using algorithms more sophisticated than open-loop control, but convolution and deconvolution can be used to analyze the behavior of closed-loop control systems as well.

Consider, for example, the basic feedback control system shown in the "Analyzing feedback loops" graphic. The process variable is sampled at regular intervals, and each measurement is subtracted from the setpoint to generate a sequence of error measurements. Those are fed into a controller that, in turn, generates a sequence of control efforts chosen to drive the process variable toward the setpoint.

The resulting process variable measurements Y will equal the results of the controller's efforts added to any uncontrollable disturbances D that might also be affecting the process. In terms of the convolution operator, that sum can be expressed as

$$Y = H * U + D$$

If the controller is also a linear process (and most are), then

PROCESS CONTROL

Deconvolution example

	16, 4, 1, 1/4, ...	← quotient (A/B)
denominator (B) → 1, -1/4	16, 0, 0, 0, ...	← numerator (A)
	16, -4, 0, 0, ...	
	0 4, 0, 0, ...	
	4, -1, 0, ...	
	0 1, 0, ...	
	1, -1/4, ...	
	0 1/4, ...	
	1/4, ...	
	0 ...	

This table shows how two sequences of sampled data values ($A = 16, 0, 0, \dots$ and $B = 1, -1/4, 0, 0, \dots$) can be deconvolved to compute a third sequence ($A/B = 16, 4, 1, 1/4, \dots$). Each row is generated using the long division algorithm except that individual "digits" can be fractions greater than 9 and less than zero, and there is no carry-over from column to column. For example, the fifth row is 4 times the denominator, and the sixth row is the difference between the fourth and fifth rows. The multiplier of 4 is chosen to produce a zero in the first column of the sixth row, and that 4 appears in the second column of the quotient.

SOURCE: Control Engineering

$$U = G * E$$

where G is the impulse response of the controller, S is the setpoint sequence and

$$E = S - Y$$

is the error sequence. These three relationships can be combined by means of simple algebra, treating convolution just like multiplication and deconvolution just like division. That is,

$$Y = G * H * (S - Y) + D$$

or

$$Y = \frac{G * H}{1 + G * H} * S + \frac{1}{1 + G * H} * D$$

This formula predicts the process variable measurements Y that will result from any given combination of setpoints S and disturbances D when a process with an open-loop impulse response of H is controlled by a feedback controller with an open-loop impulse response of G . It is the fundamental mathematical tool used by control theorists to design controllers and ana-

lyze their closed-loop behavior.

For example, the quantity

$$T_s = \frac{G * H}{1 + G * H}$$

known as the *transfer function* between the setpoint and the process variable, can be used to determine how the process variable Y will react to any given sequence of setpoints S in the absence of disturbances. That is, let $D=0$ so that

$$Y = T_s * S$$

The sequence T_s is also the impulse response of the closed-loop system. That is, T_s shows what the process variable would look like if a unity impulse were to be substituted for the setpoint while the controller is on-line controlling the process.

This phenomenon yields a test for closed-loop stability of a feedback control system. If the impulse responses of the controller and open-loop process are both known, then their convolution product $G*H$ can be used to compute T_s . If all of the digits of T_s turn out to be finite, then the closed-loop system will be stable. Otherwise, $Y = T_s * S$ will grow forever no matter what values are chosen for S , and the closed-loop system will be unstable. See the 'Stability test' graphic.

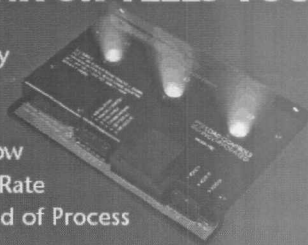
On the other hand, if the setpoint S is zero and the controller is only concerned with counteracting the effects of disturbances, then

$$Y = T_d * D$$

where the quantity

LOAD MONITOR TELLS YOU

- Mixture Viscosity
- Tool Condition
- Loss of Load
- Pump or Fan Flow
- Optimum Feed Rate
- Beginning or End of Process



UNIVERSAL POWER CELL

VERSATILE

- Works on both Fixed and Variable Frequency Power
- Even does DC and Single Phase

YOU CAN ADJUST FULL SCALE TO MATCH YOUR MOTOR

- Maximum sensitivity from small motors clear up to 150 HP

TWO ANALOG OUTPUTS

- 4-20 milliamps
- 0-10 volts

THREE BALANCED HALL EFFECT SENSORS

SIMPLE INSTALLATION

- No current transformers
- No voltage transformers

CALL NOW FOR YOUR FREE 30-DAY TRIAL
888-600-3247



LOAD CONTROLS
INCORPORATED
WWW.LOADCNTROLS.COM

PROCESS CONTROL

$$T_D = \frac{1}{1 + G * H}$$

is the transfer function between the disturbance sequence D and the process variable Y . It shows what the process variable would look like if the closed-loop system were to be stimulated with a single, unity-impulse disturbance. It too can be computed

from $G * H$, though it will typically have a different value than T_s . This is why a closed-loop system's disturbance response typically differs from its setpoint response, though if one is stable, so will the other.

The transfer functions T_s and T_D can also be used to design the controller if H is known and G has not already been select-

ed. For example, if the controller will be used primarily to track setpoint changes, T_s can be assigned a suitable value that will produce some desirable closed-loop setpoint response. The controller required to achieve that particular setpoint response would then be given by

$$G = \frac{T_s}{H * (1 - T_s)}$$

Similarly, if the controller will be used primarily to reject disturbances, then T_D can be assigned a suitable value, and the controller's impulse response can be set to

$$G = \frac{1 - T_D}{H * T_D}$$

Devising a controller that will demonstrate either of these impulse response sequences exactly can be a bit tricky, but there are mathematical techniques collectively known as *transform theory* that can help. The first step is to express G and H as fractions with numerators and denominators of finite length. For example, the infinitely long open-loop impulse response

$$H = 16, 4, 1, \frac{1}{4}, \dots$$

can also be expressed as $H=A/B$ with

$$A = 16$$

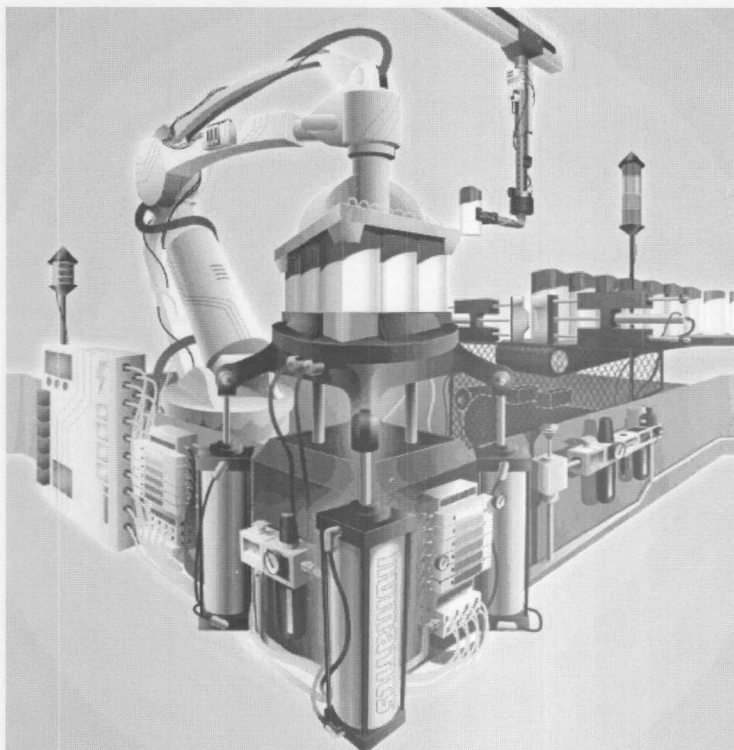
$$B = 1, -\frac{1}{4}$$

in much the same way as the infinitely long decimal value 0.14285... can be expressed as $1/7$. In fact, deconvolution and normal division are essentially the same algorithm. See the "Deconvolution example" graphic.

The simplified version of H can also be converted into a *Z transform* that can be used to analyze a closed-loop system's behavior without having to perform any convolution operations involving infinitely long sequences of sampled data. Unfortunately, it takes considerably more mathematical expertise to understand how and why *Z transforms* work.

Z transforms can be converted into *Laplace* and *Fourier transforms* by decreasing the sampling interval Δt to zero. These transforms can be used to analyze the behavior of analog control systems where data is processed in a continuous stream rather than at regular intervals. **ce**

Vance VanDoren (controleng@msn.com) is consulting editor to Control Engineering.



Innovate. Optimize. Achieve.

From idea to implementation, leading engineers choose Numatics as their single source for quality components, technologically-advanced design resources, and quick response time in delivery and service from anywhere around the world. Visit www.numatics.com today to learn more about our complete product line and let us introduce you to the world of fluid automation.



numatics
Moving Industry

www.numatics.com